Amendments to the Claims:

This listing of claims will replace all prior versions, and listings of claims in the application:

Listing of Claims:

Claim 1. (currently amended) A method of operating a programmable logic integrated circuit comprising:

loading an initial value in a count register that is a part of a watchdog timer circuit ~~on~~ integrated as part of a ~~the~~ programmable logic integrated circuit disposed on a single die;

clocking the count register to advance the count register to a next value with each clock;

periodically reloading the count register with the initial value, wherein the reloading is caused by receiving a first magic value that configures the watchdog timer circuit to respond to a second magic value that is different from the first magic value, wherein after receiving the first magic value, upon receiving the second magic value, resetting the watchdog timer circuit to the initial value;

when the stored count value held in the count register that is a part of the watchdog timer circuit reaches a final value, asserting a triggered signal output; and

upon receiving the triggered signal output in a reset logic block on the programmable logic integrated circuit, causing reloading of configuration data from an external source into the programmable logic integrated circuit.

Claim 2. (original) The method of claim 1 wherein the external source is a nonvolatile memory.

Claim 3. (original) The method of claim 1 wherein the external source is a serial EPROM.

Claim 4. (previously presented) The method of claim 1 wherein the final value causes an overflow condition for the count register that is a part of the watchdog timer circuit.

Claim 5. (original) The method of claim 1 wherein the watchdog timer circuit increments the stored count value at each clock pulse.

Claim 6. (original) The method of claim 1 wherein the watchdog timer circuit decrements the stored count value at each clock pulse.

Claim 7. (currently amended) The method of claim 1 wherein periodically reloading the count register comprises:

writing a the first magic value into a reload register that is a part of the watchdog timer circuit; and

when the first magic value is received in the reload register, resetting the count register that is a part of the watchdog timer circuit to the initial value.

Claims 8. – 9. (canceled)

Claim 10. (previously presented) The method of claim 1 further comprising:

using the configuration data to configure an embedded processor portion and a programmable logic portion on the programmable logic integrated circuit.

Claim 11. (original) The method of claim 1 wherein to avoid asserting the triggered signal output, a periodic reload of the watchdog timer circuit should be performed during a time period it takes the watchdog timer circuit to count from the initial value to the fina value.

Claim 12. (original) The method of claim 11 wherein the period is less than abou t two minutes.

Claim 13. (original) The method of claim 11 wherein the time period depends or clock frequency used to clock the watchdog timer circuit.

Claim 14. (original) The method of claim 1 wherein the initial value is 0 and the final value is a maximum count value permitted by the count register.

Claim 15. (original) The method of claim 1 wherein the count register comprises 32 bits.

Claims 16. - 43. (cancelled).

Claim 44. (currently amended) A method of operating a programmable logic integrated circuit comprising:

clocking a watchdog timer circuit on the programmable logic integrated circuit to advance a count register that is a part of the watchdog timer circuit, wherein the programmable logic integrated circuit and the watchdog timer circuit are disposed on the same die;

loading a first magic value into a reload register that is a part of the watchdog timer circuit, which resets the count register to an initial value, wherein the first magic value configures the watchdog timer circuit to respond to a second magic value that is different from the first magic value;

after loading the first magic value, loading a the second magic value into the reload register, which causes the count register to reset the initial value;

after loading the first magic value into the reload register, loading a value other than the second magic value into the reload register, which causes the watchdog timer circuit to generate a triggered signal; and

receiving the triggered signal in a reset logic block on the programmable logic integrated circuit, which causes a reloading of configuration data from an external source into the programmable logic integrated circuit.

Claim 45. (cancelled)

Claim 46. (previously presented) The method of claim 44 wherein the configuration data is used to configure an embedded processor portion and a programmable logic portion on the programmable logic integrated circuit.

Claim 47. (previously presented) The method of claim 44 wherein the watchdog timer circuit is located in an embedded processor portion and the reset logic block is located in a programmable logic portion on the programmable logic integrated circuit.

Claim 48. (previously presented) The method of claim 44 further comprising:

allowing the count register that is a part of the watchdog timer circuit to advance to a final value before the first or second magic values are loaded, which causes the watchdog timer circuit to generate the triggered signal.

Claim 49. (original) The method of claim 44 wherein the initial value is 0.

Claim 50. (original) The method of claim 44 wherein the initial value is a value other than 0.

Claim 51. (currently amended) The method of claim 44 wherein the ~~first magic value is different from the~~ second magic value configures the watchdog timer circuit to respond to a third magic value that is different from the second magic value.

Claim 52. (original) The method of claim 48 wherein the final value is user-selectable.

Claim 53. (original) The method of claim 48 wherein the final value is the maximum count permitted by the count register.

Claim 54. (original) The method of claim 44 wherein in a debug mode, the count register does not advance.